

# 2进程描述、控制

2018年10月8日 8:00

◆

## ◆ 前趋图和程序执行

### 1. 操作系统四大特征都是基于进程而形成的

#### 一. 前趋图

##### 1. 前趋图Precedence Graph

###### 1) 有向无循环图Directed Acyclic Graph

2) 每个结点可用于描述一个程序段或进程, 乃至一条语句

3) 有向边表示两个结点之间存在的偏序关系(Partial Order) 或前趋关系(Precedence Relation)“ $\rightarrow$ ”

4)  $(P_i, P_j) \in \rightarrow$ , 可写成  $P_i \rightarrow P_j$

##### 2. 结点

(1) 把没有前趋的结点称为初始结点(Initial Node)

(2) 把没有后继的结点称为终止结点(Final Node)

1) 每个结点有一个重量(Weight), 表示该结点所含有的程序量或执行时间

#### 二. 程序顺序执行

1. 顺序执行: 仅当前一操作(程序段)执行完后, 才能执行后继操作

##### 2. 程序顺序执行时的特征

(1) 顺序性: 处理机严格按程序规定顺序执行操作: 每一操作必须在上一操作结束后开始

(2) 封闭性: 程序运行时独占全机资源, 资源的状态(除初始状态外)只有本程序才能改变。程序一旦开始执行, 其执行结果不受外界因素影响

(3) 可再现性: 只要程序执行时的环境和初始条件相同, 当程序重复执行时, 不论它是从头到尾不停顿地执行, 还是“停停走走”地执行, 都将获得相同的结果

#### 三. 程序并发执行

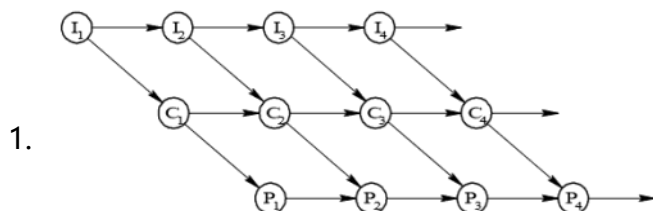


图 2-3 并发执行时的前趋图

1) 在该例中存在下述前趋关系:  $I_i \rightarrow C_i$ ,  $I_i \rightarrow I_{i+1}$ ,  $C_i \rightarrow P_i$ ,  $C_i \rightarrow C_{i+1}$ ,  $P_i \rightarrow P_{i+1}$

2) 而  $I_{i+1}$  和  $C_i$  及  $P_{i-1}$  是重迭的, 在  $P_{i-1}$  和  $C_i$  以及  $I_{i+1}$  之间, 可以并发执行

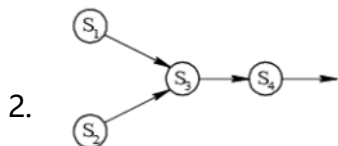


图 2-4 四条语句的前趋关系

1)  $S_3$  必须在  $a$  和  $b$  被赋值后方能执行

2)  $S_4$  必须在  $S_3$  之后执行

3)  $S_1$  和  $S_2$  则可以并发执行, 因为它们彼此互不依赖

##### 3. 程序并发执行时的特征

1) 间断性: 并发执行的程序之间, 有相互制约的关系, 导致“执行—暂停—执行”的活动规律

2) 失去封闭性: 共享资源的状态将由多个程序来改变。这样, 某程序在执行时, 必然会受到其它程序的影响。例如, 当处理机这一资源已被某个程序占有时, 另一程序必须等待。

- 3) 不可再现性: 由于失去了封闭性, 执行时的环境和初始条件相同, 但得到的结果却不相同



◆ 进程的描述

## 一. 进程的定义和特征

### 1. 进程控制块(Process Control Block)

- 1) 为使程序(含数据)能独立运行, 应为之配置一进程控制块
- 2) 进程实体/进程映像: 由程序段、相关的数据段和 PCB 三部分构成
- 3) 创建进程实质上是创建PCB; 而撤消进程实质上是撤消PCB

### 2. 进程的定义

- 1) 进程是程序的一次执行
- 2) 进程是一个程序及其数据在处理机上顺序执行时所发生的活动
- 3) 进程是程序在一个数据集合上运行的过程, 它是系统进行资源分配和调度的一个独立单位
- 4) 进程是进程实体的运行过程, 是系统进行资源分配和调度的独立单位

### 3. 进程的特征

#### 1) 动态性

- (1) 进程的实质是进程实体的一次执行过程, 因此, 动态性是进程的最基本的特征
- (2) 由创建而产生, 由调度而执行, 由撤消而消亡
- (3) 进程实体有生命期, 而程序则只是一组存放于某种介质上有序指令的集合, 是静态的

#### 2) 并发性: 多个进程实体同存于内存中, 且能在一段时间内同时运行

#### 3) 独立性: 独立运行、独立分配资源和独立接受调度

#### 4) 异步性: 进程按各自独立的、不可预知的速度向前推进, 或说进程实体按异步方式运行

- (1) 应通过配置相应的进程同步机制, 保证进程并发执行的结果是可再现的

#### 1. 进程和作业

- 1) 进程总在内存, 作业刚提交时在外存等待
- 2) 作业可由多个进程组成, 反之不然
- 3) 作业的概念主要出现在批处理系统中, 进程用在几乎所有多道系统中

#### 2. 进程和程序

- 1) 进程动态, 程序静态
- 2) 程序可对应多个进程, 进程可包含多个程序

## 二. 进程的基本状态及转换

### 1. 进程的三种基本状态

#### 1) 就绪(Ready)状态

- (1) 已分配到除 CPU 以外的必要资源, 只要获得 CPU, 便可立即执行
- (2) 通常将它们按一定策略排成一个就绪队列

#### 2) 执行(Running)状态

- (1) 进程已获得 CPU, 其程序正在执行
- (2) 处理机数量决定处于执行状态的进程的数量

#### 3) 阻塞(Block)状态/等待状态/封锁状态

- (1) 发生某事件而暂时无法继续执行时, 放弃处理机而暂停
- (2) 此时引起进程调度, OS把处理机分配给另一个就绪程序
- (3) 根据请求 I/O, 申请缓冲空间等阻塞原因把处于阻塞状态的进程排成多个队列

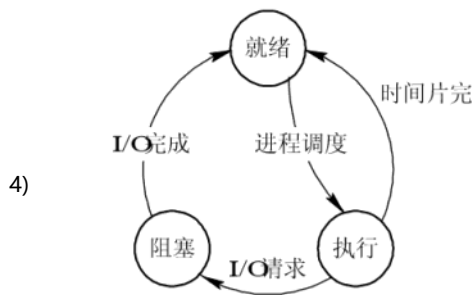


图 2-5 进程的三种基本状态及其转换

## 2. 创建状态和终止状态

### 1) 创建状态

- (1) 创建过程：创建 PCB，并填写必要的管理信息；分配运行时必要的资源；把该进程转入就绪状态并插入就绪队列之中
- (2) 1若已创建PCB，而资源尚未分配完，此时的状态即为~
- (3) 可以根据系统性能或主存容量的限制，推迟创建状态进程的提交

### 2) 终止状态

- (1) 当一个进程到达了自然结束点，或出现了无法克服的错误，或被操作系统所终结，或被其他有终止权的进程所终结，将进入~
- (2) 终止过程：善后处理，将其 PCB 清零，并将 PCB 空间返还系统
- (3) 进入终止态以后不能再执行，但在操作系统中依然保留状态码和一些计时统计数据，供其它进程收集，一旦其它进程完成了对终止状态进程的信息提取，操作系统将删除该进程



图 2-7 进程的五种基本状态及转换

## 三. 挂起操作和进程状态的转换

### 1. 挂起操作的引入原因

- 1) 终端用户的请求：终端用户在程序运行期间发现有可疑问题时，希望程序暂停执行，以使用户研究其执行情况或对程序进行修改
- 2) 父进程请求：考查和修改该子进程，或者协调各子进程间的活动
- 3) 负荷调节的需要：系统把一些不重要的进程挂起，以保证正常运行
- 4) 操作系统的需要：检查运行中的资源使用情况或进行记账

### 2. 进程状态的转换

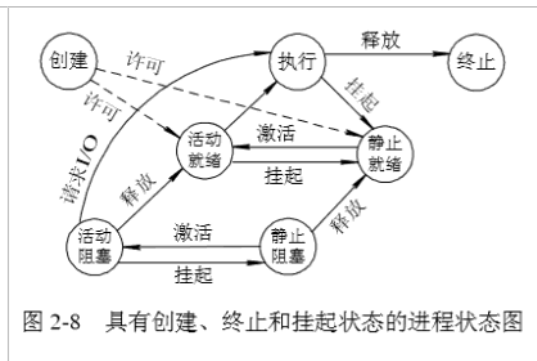
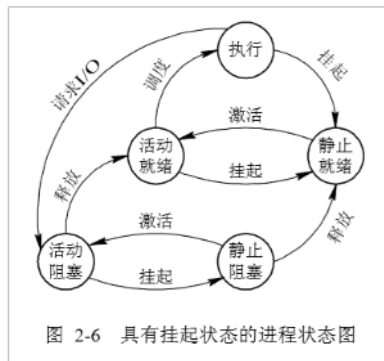
#### 1) 活动就绪→静止就绪

- (1) 未被挂起的就绪状态，称为活动就绪，表示为 Readya
- (2) 用挂起原语 Suspend 将该进程挂起后，该进程便转变为静止就绪状态，表示为 Readys

#### 2) 活动阻塞→静止阻塞

- (1) 未被挂起的阻塞状态，称活动阻塞状态，表示为 Blocked<sub>a</sub>
- (2) 用 Suspend 原语将它挂起后，进程便转变为静止阻塞状态，表示为 Blocked<sub>s</sub>。处于该状态的进程在其所期待的事件出现后，将从静止阻塞变为静止就绪

- 3) 静止就绪→活动就绪：处于 Readys 状态的进程，用激活原语 Active 激活后，转为 Readya 状态
- 4) 静止阻塞→活动阻塞：处于 Blockeds 状态的进程，用激活原语 Active 激活后，转为 Blockeda 状态



#### 四. 进程管理中的数据结构

##### 1. 操作系统中用于管理控制的数据结构

- 1) 资源信息表/进程信息表：标识、描述、状态等信息和一些指针
- 2) 内存、设备、文件、进程都有表，其中进程表又被称为进程控制块

##### 2. 进程控制块的作用

- 1) 作为独立运行基本单位的标识
- 2) 实现间断性运行的方式（保存cpu现场信息）
- 3) 提供进程管理所需的信息（保存数据指针、文件设备等资源的访问、资源清单）
- 4) 提供进程调度所需的信息（保存状态信息、优先级、等待时间、执行时间等）
- 5) 实现与其他进程的同步与通信（实现进程通信的区域或通信队列指针）

##### 3. 进程控制块中的信息

###### 1) 进程标识符

- (1) 内部标识符：为方便调用每一个进程赋予一个惟一的数字标识符，通常是一个序号
- (2) 外部标识符：由创建者提供，通常是字母、数字组成，往往是在访问该进程时使用
  - i. 为了描述进程的家族关系，还应设置父进程标识及子进程标识
  - ii. 还可设置用户标识，以指示拥有该进程的用户

###### 2) 处理机状态/处理机上下文

- (1) 通用寄存器/用户可视寄存器：它们是用户程序可以访问的，用于暂存信息
- (2) 指令计数器：其中存放了要访问的下一条指令的地址
- (3) 程序状态字 PSW：其中含有状态信息，如条件码、执行方式、中断屏蔽标志等
- (4) 用户栈指针：存放过程和系统调用参数及调用地址的栈的指针，指向该栈的栈顶

###### 3) 进程调度信息

- (1) 进程状态：指明进程的当前状态，作为进程调度和对换时的依据
- (2) 进程优先级：一个整数，优先级高的进程应优先获得处理机
- (3) 其它信息：与进程调度算法有关，如，等待 CPU 的时间总和、已执行时间总和
- (4) 阻塞原因：进程由执行状态转变为阻塞状态所等待发生的事件

###### 4) 进程控制信息

- (1) 程序和数据地址：程序和数据所在地址，以便从 PCB 中找到其程序和数据
- (2) 进程同步和通信机制：消息队列指针、信号量等
- (3) 资源清单：除 CPU 以外的、进程所需的全部资源及已经分配到该进程的资源的清单
- (4) 链接指针：给出本进程(PCB)所在队列中的下一个进程的 PCB 的首地址

##### 4. 进程控制块的组织方式

- 1) 线性方式：所有PCB组织在一张线性表。只适合进程数目不多的系统
- 2) 链接方式：把同状态的 PCB用链接字链接成就绪队列、若干阻塞队列和空白队列等

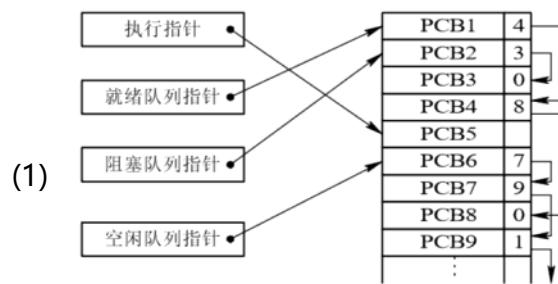


图 2-9 PCB 链接队列示意图

- 3) **索引方式**:根据所有进程的状态建立几张索引表, 并把各索引表在内存的首地址记录在内存的一些专用单元中。在每个索引表的表目中, 记录具有相应状态的某个 PCB 在 PCB 表中的地址

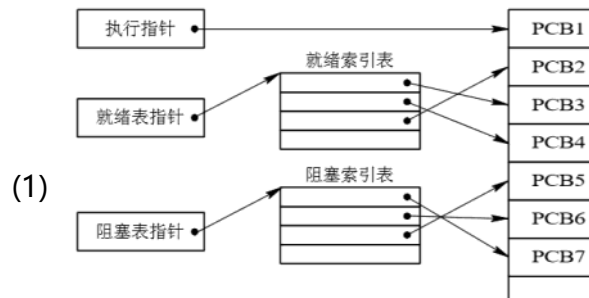


图 2-10 按索引方式组织 PCB

- ◆
- ◆ 进程控制

## 一. 操作系统内核

- 1) 通常与硬件紧密相关的模块、常用设备的驱动程序、运行频率较高的模块都安排在紧靠硬件的软件层次中, 将他们常驻内存, 称其**OS内核**
- 2) 处理机系统态/管态/内核态: 有较高特权, 能执行一切指令, 访问所有寄存器和存储区
- 3) 用户态/目态: 仅能执行规定指令, 访问指定寄存器和存储区, 是一般情况的执行状态

### 1. 支撑功能

- 1) **中断处理**: 最基本的功能, 系统调用、键盘命令的输入、进程调度、设备驱动等都需要它
- 2) **时钟管理**: 如时间片结束后, 时钟管理会产生一个中断信号
- 3) **原语操作**: 链表操作、进程同步等
  - (1) 原语Primitive: 若干条指令组成的, 有一定功能的一个过程
    - i. 是原子操作Action Operation: 不可分割
    - ii. 要么全做, 要么不做, 不能被中段
    - iii. 在系统态下执行, 常驻内存

### 2. 资源管理功能

- 1) **进程管理**: 调度分派、创建撤销、同步通信等
- 2) **存储器管理**: 逻辑/物理地址转换、内存分配回收、内存保护对换等
- 3) **设备管理**: 设备驱动程序、缓冲管理、设备分配和独立性功能模块等

## 二. 进程的创建

1. 进程的层次结构: 父进程、子进程、孙进程组成的进程家族
2. 进程图Process Graph

- 1) 进程图是描述进程家族关系的有向树
- 2) 结点(圆圈)代表进程
- 3) 进程 D 创建了进程 I, 称 D 是 I 的父进程Parent Process, I 是 D 的子进程Progeny Process

- 4) 创建父进程的进程是祖先进程，树的根节点是进程家族的祖先Ancestor
- 5) 由父进程指向子进程的有向边描述了父子关系
- 6) 子进程可以继承父进程的资源，如，父进程打开的文件，父进程所分配到的缓冲区等
- 7) 子进程被撤消时，应将继承的资源归还给父进程。撤消父进程时，应同时撤消其所有的子进程

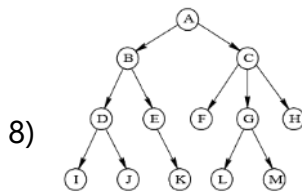


图 2-11 进程树

### 3. 引起创建进程的事件

#### 1) 系统内核：

- (1) 用户登录：分时系统合法用户登陆后，该终端得到一个进程，并被插入就绪队列中
- (2) 作业调度：批处理系统调度到某作业时，将为该作业创建进程，再插入就绪队列中
- (3) 提供服务：运行中的用户程序提出某请求后，系统将专门创建一个进程来提供服务

#### 2) 应用请求：进程自己创建一个新进程，以便以并发运行方式完成特定任务，如IO

### 4. 进程的创建(Creation of Process) 即原语 Creat()内容：

- 1) 申请空白 PCB：为新进程申请获得惟一的数字标识符，并从 PCB 集合中索取一个空白 PCB
- 2) 为新进程分配资源：为新进程的程序和数据以及用户栈分配必要的内存空间
- 3) 初始化进程控制块
  - (1) 初始化标识信息，将系统分配的标识符和父进程标识符填入新 PCB 中
  - (2) 初始化处理机状态信息，使程序计数器指向程序的入口地址，使栈指针指向栈顶
  - (3) 初始化处理机控制信息，将进程的状态设置为就绪状态或静止就绪状态，对于优先级，通常是将它设置为最低优先级，除非用户以显式方式提出高优先级要求
  - (4) 将新进程插入就绪队列，如果就绪队列能够接纳新进程，便将新进程插入

## 三. 进程的终止

### 1. 引起进程终止的事件

- 1) 正常结束：用于表示进程已经运行完成的指示。批处理系统中，通常在程序的最后安排一条 Halt 指令或终止的系统调用。在分时系统中，用户可利用 Logs off 去表示进程运行完毕
- 2) 异常结束：某些错误和故障迫使进程终止(Termination of Process)
  - (1) 越界错误：程序所访问的存储区已越出该进程的区域
  - (2) 保护错：进程试图去访问一个不允许访问的资源或文件，或者以不适当的方式进行访问，如，写一个只读文件
  - (3) 非法指令：试图去执行不存在的指令。可能是程序错误地转移到数据区，把数据当成了指令
  - (4) 特权指令错：用户进程试图去执行一条只允许 OS 执行的指令
  - (5) 运行超时：进程的执行时间超过了指定的最大
  - (6) 等待超时：进程等待某事件的时间超过了规定的最大值
  - (7) 算术运算错：进程试图去执行一个被禁止的运算，如被 0 除
  - (8) I/O 故障：I/O 过程中发生了错误等
- 3) 外界干预：进程应外界的请求而终止运行
  - (1) 操作员或操作系统干预：如，发生了死锁，由操作员或操作系统 终止该进程
  - (2) 父进程请求：父进程有终止子孙进程的权力。父进程提出请求时，系统将终止该进程

(3) 父进程终止：当父进程终止时，OS 也将它的所有子孙进程终止

## 2. 进程的终止过程

- 1) 读出进程状态（根据标识符从 PCB 集合中检索其 PCB）
- 2) 终止执行，并置调度标志为真，用于指示该进程应重新调度
- 3) 终止所有子孙进程，以防它们成为不可控的进程
- 4) 资源归还给其父进程或系统
- 5) 移出 PCB（从所在队列(或链表)），等待其他程序来搜集信息

## 3. 进程的阻塞与唤醒

- 1) 引起进程阻塞和唤醒的事件：
  - (1) 请求系统资源失败：如打印机，仅在其他进程在释放出打印机的同时，才将请求进程唤醒
  - (2) 等待某种操作：如进程在启动了 I/O 操作后，便自动进入阻塞状态，在 I/O 操作完成后，再由中断处理程序或中断进程将该进程唤醒
  - (3) 新数据尚未到达：如 A 尚未将数据输入完毕，则进程 B 将因没有所需的处理数据而阻塞；一旦进程 A 把数据输入完毕，便可去唤醒进程 B
  - (4) 无新任务到达：如，系统中的发送进程，已有的数据已全部发送完成而又无新的发送请求，这时发送进程将使自己进入阻塞状态；仅当有进程提出新的发送请求时，才唤醒
- 2) 进程阻塞过程：调用阻塞原语 block 把自己阻塞。可见，阻塞是进程自身的一种主动行为
  - (1) 先立即停止执行，把进程控制块中的现行状态由“执行”改为“阻塞”
  - (2) 将 PCB 插入到具有相同事件的阻塞(等待)队列
  - (3) 转调度程序进行重新调度，即，在 PCB 中保留被阻塞进程的处理机状态，再按新进程的 PCB 中的处理机状态设置 CPU 的环境
- 3) 进程唤醒过程：调用唤醒原语 wakeup( )
  - (1) 把进程从等待该事件的阻塞队列中移出
  - (2) 将其 PCB 中的现行状态由阻塞改为就绪
  - (3) 将该 PCB 插入到就绪队列中

## 四. 进程的挂起与激活

1. 进程的挂起：利用挂起原语 suspend( )将指定进程或处于阻塞状态的进程挂起
  - 1) 检查进的状态，活动就绪状态，改为静止就绪；活动阻塞状态改为静止阻塞
  - 2) 把该进程的 PCB 复制到某指定的内存区域，方便用户或父进程考查该进程的运行情况
  - 3) 若被挂起的进程正在执行，则转向调度程序重新调度
2. 进程的激活过程：利用激活原语 active( )
  - 1) 将进程从外存调入内存
  - 2) 静止就绪改为活动就绪；静止阻塞改为活动阻塞
    - (1) 插入就绪队列时假如采用的是抢占调度策略，应由调度程序进行优先级的比较，如果被激活进程的优先级高，立即剥夺当前进程的运行，把处理机分配给刚被激活的进程

**进程不可能由自己创建/唤醒/激活/终止。可能由自己挂起/阻塞**

- i.
- ii.
- iii.
- iv.
- v.
- vi.
- vii. -----我是底线-----